Overview:
We will be making an app that will be able to collect responses from different devices where all the responses would be saved to a cloud and synced between the devices. This App will demonstrate how we can crowdsource responses to a question. At the end we will use this app as a trivia app to collect your responses to some questions. The app interface would look like the photo to your right.

1. Making a new project and using App Inventor:
   b. On the upper left side you will see the word “Projects”. Press on it and then choose new project as in the photo below.
   c. Choose a name for your project. (CloudPoll for example)
   d. (optional) **Connect your Android phone/tablet** to your project to test while you build.

First, Designing The User Interface:
We will need a box to enter the name of the user and two buttons, which will represent two options (A and B for our program). We will also need to display the names of the people who chose each option.

The Name Box:
1. From the Layout section in the Palette choose “HorizontalArrangement” and drag it to the screen. Change the following properties:
   a. Set the width to “Fill parent”.
   b. Set AlignVertical to Center.
2. From the User Interface section in the Palette drag a Label and then a TextBox to the “HorizontalArrangement” box on the screen.
   a. Change the label text in properties to “Name: “.
   b. Set the width for the TextBox to “Fill parent”.
   c. Rename the text box to VoterName.

The Options Buttons:
1. Go to the User Interface menu and drag two buttons into the screen.
2. Change their text properties to A and B.
3. Rename the buttons to “OptionA” and “OptionB”.

The list of people choices:
1. Drag two labels from the User Interface menu into the screen and replace their text (from properties) to something like “A: None” and “B: None”.
2. Rename the labels to “Avoters” and “Bvoters”. Your screen should look like this:
Second, Designing The Blocks: Here we will be programming and defining the logic of our App.

CloudDB: This is the component we will use to collect all the responses on the same database.

- From the Experimental section in the Palette drag a CloudDB component into the screen.

The Blocks: Now that we have all the necessary components on our screen, we can start to program them using the blocks. Press on the Blocks button on the right upper corner.

1. Initializing the screen:
   a. Press on Screen in the blocks menu to choose from the blocks that controls the screen functions. Look for “when screen1. Initialize” and drag it to the whiteboard.

   ![Diagram](image)

   b. From the CloudDB blocks grap the block you see on the right and put it in the screen1 initializing block.
   c. Attach a text box to the tag place and set the text to “A”
   d. Attach an empty list (From the lists menu) to “valueIfTagNotThere”.
   e. Do steps b to d again but change the tag text to “B”.

   You should have the following blocks structure:

   ![Diagram](image)

2. Button Clicks: we need to modify our database each time a click happens to the buttons A or B.
a. From “OptionA” blocks choose “When OptionA.Click”.
b. From “CloudDB” Blocks drag the “Call CloudDB1.AppendValueToList” Block and attach it the previous block.
c. Attach an empty text box to the tag and fill it with the tag name, which is “A” in this case.
d. From the VoterName blocks drag the “VoterName.Text” block and attach it to “ItemToAdd”.

It should look like this:

![Diagram of block flow]

e. Do steps a to d again but for the B option.

3. Displaying the new statistic on the interface buttons and labels.
   a. Go to the CloudDB1 blocks and drag the “When CloudDB1.DataChanged” block.
   b. From the Control Blocks drag an “if, then” block. Make sure to add an extra “else if” block to it. As in the following picture:

   ![Diagram of if-then-else flow]

   c. Attach the text comparing block from the text block to the if statement. Then, from the “When CloudDB1.DataChanged” block, press on tag and drag the get tag block to the text comparing block. In the other
empty space of the text comparing block put a text box and fill it with the tag name we are trying to get stats for. Then do the same thing for the B tag and attach it to the “else if” statement. The blocks should look like this:

d. We will be changing the button text to display the number of people who chose it and the labels in the lower part of the screen to display the list of the names of people who chose either option. (No step here!)

e. Drag the “Set OptionA.Text to” block from the OptionA blocks.
   i. Connect it to the “then” block space.
   ii. Use a Join block from the text blocks.
   iii. Connect an empty text box to the first space of the Join block and set it to “A: “.
   iv. In the second space put a “length of list” block from the Lists blocks and connect it to “get value”, which you can get from the “When CloudDB1.DataChanged” block by pressing on the value icon below it.

f. Drag “Set AVoters Text to” block from the AVoters blocks. Then, do the same steps in e with one small change. Instead of putting “length of list”, just connect the “get value block” directly.

It should look like this:
g. Do the steps e and f again but for the B option.

Your App should be ready to work now!!

This is how all the blocks should look in the end.